# On the Power of Relaxed Local Decoding Algorithms*

Tom Gur†
University of Warwick
tom.gur@warwick.ac.uk

Oded Lachish
Birkbeck, University of London
oded@dcs.bbk.ac.uk

**Abstract**

A *locally decodable code* (LDC) $C\colon \{0,1\}^k \to \{0,1\}^n$ is an error correcting code that admits algorithms for recovering individual bits of the message by only querying a few bits of a noisy codeword. LDCs found a myriad of applications both in theory and in practice, ranging from probabilistically checkable proofs to distributed storage. However, despite nearly two decades of extensive study, the best known constructions of LDCs with $O(1)$-query decoding algorithms have super-polynomial blocklength.

The notion of *relaxed LDCs* is a natural relaxation of LDCs, which aims to bypass the foregoing barrier by requiring local decoding of nearly all individual message bits, yet allowing decoding failure (but not error) on the rest. State of the art constructions of $O(1)$-query relaxed LDCs achieve blocklength $n = O\left(k^{1+\gamma}\right)$ for an arbitrarily small constant $\gamma$.

Using algorithmic and combinatorial techniques, we prove an impossibility result, showing that codes with blocklength $n = k^{1+o(1)}$ cannot be relaxed decoded with $O(1)$-query algorithms. This resolves an open problem raised by Goldreich in 2004.

*Extended abstract of this work appears at SODA 2020.
†Tom Gur is supported by the UKRI Future Leaders Fellowship MR/S031545/1.

# Contents

# 1 Introduction

Locally decodable codes (LDC) are fundamental objects in algorithmic coding theory. Loosely speaking, an LDC is an error correcting code with a robust local-to-global structure which admits a randomised algorithm that can recover individual message bits by probing a minuscule portion of a noisy codeword. Thus, rather than reading the entire codeword to decode the entire message, an LDC allows algorithms that read a small number of locations to decode a single bit of the message.

More precisely, we consider codes $C\colon \{0,1\}^k \to \{0,1\}^n$ with linear distance, and say that a code $C$ is an LDC if there exists a randomised algorithm, called a *local decoder*, that is given a location $i \in [k]$ and query access to an input $w \in \{0,1\}^n$ such that if $w$ is sufficiently close (typically within distance that is proportional to the distance of the code) to a valid codeword $C(x)$, the decoder outputs $x_i$ with high probability. The maximal number of queries that the decoder makes is called the *locality* of the code.

Since the systematic study of LDCs was initiated in the seminal work of Katz and Trevisan [KT00], these codes received much attention and made a profound impact on algorithms, cryptography, complexity theory, program checking, data structures, quantum computing, pseudorandomness, and other areas in theoretical computer science (see surveys [Tre04, Yek12, KS17] and references therein), as well as led to significant practical applications in distributed storage [HSX+12].

Unfortunately, despite the success and attention that LDCs gained in the last two decades, there remains a chasm between the best known algorithms and impossibility results. Specifically, the best general lower bounds that are currently known (cf. [KdW04, Woo12], building on [KT00]), show that any $\ell$-local LDC must have blocklength

$$n = \Omega\left(k^{1+\frac{1}{\ell-1}}\right),$$

where throughout, $k$ is the dimension of the code. Moreover, for very specific regimes of parameters, improvements are known, e.g., for 2-local LDCs [KdW04]. In stark contrast, the state-of-the-art construction of $O(1)$-local LDCs has a *super-polynomial* blocklength (cf. [Efr12], building on [Yek08]).

The foregoing algorithmic barrier has led to the study of *relaxed locally decodable codes*, in short "relaxed LDCs", which were introduced in the highly influential work of Ben-Sasson, Goldreich, Harsha, Sudan, and Vadhan [BGH+04]. In a recent line of research [GGK15, GG16, BDG17, GG18, GR18, GRR18, BGGZ18, CGS20] relaxed LDCs and their variants (such as relaxed locally correctable codes) have been studied and used to obtain applications to PCPs [MR10, DH13, RR19], property testing [CG18], data structures [CGdW09], and probabilistic proof systems [GGK15, GG16, GR17, GR18].

Loosely speaking, this relaxation of LDCs allows the local decoder to declare "decoding failure" on a small fraction of the indices, yet crucially, still avoid errors. More accurately, a *relaxed LDC* $C\colon \{0,1\}^k \to \{0,1\}^n$ is a code (with linear distance) for which there exists a *decoding radius* $\delta$ (typically proportional to the relative distance of the code) and a randomised algorithm, called the *relaxed local decoder* that receives an index $i \in [k]$ and oracle access to a string $w \in \{0,1\}^n$ that is $\delta$-close to a codeword $C(x)$. The relaxed local decoder is allowed to make a small number of queries to $w$ (typically $O(1)$ queries) and is required to satisfy the following conditions:

1. *Completeness:* If the input is a valid codeword (i.e., $w = C(x)$), the relaxed local decoder must always output $x_i$.

2. *Relaxed decoding:* Otherwise, with high probability, the decoder must either output $x_i$ or a special "reject" symbol $\perp$ (indicating the decoder detected an error and is unable to decode).

As observed in [BGH+04], the foregoing two conditions suffice for obtaining a third condition, which guarantees that the relaxed local decoder may only reject (i.e., output $\perp$) on an arbitrarily small fraction of the coordinates. (See Section 3 for a formal definition of relaxed LDCs, covering all three conditions.)

This seemingly modest relaxation turns out to allow the usage of extremely powerful tools from the theory of probabilistically checkable proofs (PCPs) to design highly-efficient randomised algorithms. Relying on the notion of PCPs of proximity, which they also introduced and constructed, Ben-Sasson et al. [BGH+04] constructed a relaxed LDC with *nearly-linear length*. More precisely, they showed that for every constant $\gamma > 0$ there exists an $O(1)$-local relaxed LDC $C\colon \{0,1\}^k \to \{0,1\}^n$ with nearly-linear blocklength $n = k^{1+\gamma}$. We remark that 15 years later, there is no known construction of relaxed LDCs that improves on [BGH+04].

While the aforementioned relaxed LDCs have blocklength that is nearly exponentially shorter than that of any known (non-relaxed, i.e., standard) LDC, they do *not* break the currently known lower bound on non-relaxed LDCs (cf. [KT00]). This led Goldreich [Gol04] to raise the following open problem:

Do there exist $O(1)$-local relaxed decoding algorithms for codes with blocklength $n = k^{1+o(1)}$?

### 1.1 Our results

Our main contribution resolves the foregoing open problem by providing a strong negative answer. Namely, we prove the following theorem, which shows that $O(1)$-local relaxed LDCs cannot achieve blocklength $n = k^{1+o(1)}$.

**Theorem 1** (informal, see Theorem 4.1)**.** *For any $\ell \in \mathbb{N}$ and $\delta \in (0,1)$, there exists a constant $\alpha = \alpha(\ell,\delta) > 0$ such that every $\ell$-local relaxed LDC $C\colon \{0,1\}^k \to \{0,1\}^n$ with decoding radius $\delta$ satisfies $n = \Omega\left(k^{1+\alpha}\right)$.*

To the best of our knowledge, this is the first non-trivial lower bound that was shown for *relaxed* LDCs. We remark that Theorem 1 directly extends to the setting of linear *relaxed locally correctable codes*, recently introduced in [GRR18].

Our lower bound is proved via *algorithmic and combinatorial techniques*; namely, it can be viewed as showing that any $\ell$-local relaxed LDC is recoverable with high probability from the binary erasure channel (BEC), which is analysed via relaxed sunflower lemmas. See Section 2.

Our results leave several intriguing open problems, which we highlight next.

**The role of adaptivity.** For relaxed LDCs with constant decoding radius and non-adaptive decoders (i.e., where each query is made independently of the answers to previous queries), the parameter $\alpha$ in our lower bound of $n = \Omega\left(k^{1+\alpha}\right)$ takes the form of $\alpha = \frac{1}{O(\ell^2)}$. This dependency is quite close to that in the upper bound of Ben-Sasson et al. [BGH+04], who showed that for any $\ell \in \mathbb{N}$ there exists an $\ell$-local non-adaptive relaxed LDC $C\colon \{0,1\}^k \to \{0,1\}^n$ with blocklength $n = k^{1+\Theta(1/\sqrt{\ell})}$. It is both natural and appealing to understand whether such a lower bound also holds for adaptive decoders, and we leave this as an open problem.

**Stronger bounds for $\ell = 2$.** While there are known exponential lower bounds on the blocklength of 2-local LDCs over binary alphabet [Oba02, KdW04], no such result is known for relaxed LDC.

We conjecture that exponential lower bounds can be shown for relaxed LDCs as well, and leave this problem to future work.

**Larger alphabet size.** We remark that our techniques naturally generalise to the setting of codes with a larger (non-binary), yet *constant*, alphabet size. It is not clear, however, whether our lower bound extends to the setting of codes with *super-constant* alphabet size. We raise this as an open problem.

### 1.2 Related works

There is an extensive literature that is concerned with lower bounds on (non-relaxed) locally decodable codes in various regimes (see, e.g., [KT00, DJK$^+$02, GKST02, Oba02, KdW04, WdW05, Woo12]), as well as for the closely related notion of locally correctable codes (see, e.g., [BDYW11, BDSS11, BGT16, DSW17]), in which the goal is to correct a bit of the codeword rather than a bit of the message. We stress that none of the aforementioned bounds apply for *relaxed LDC* (see discussion in Section 2.1).

Another related notion is that of *locally testable codes* [GS06], which are, loosely speaking, codes for which there exist a probabilistic algorithm that accepts valid codewords, and rejects inputs that are "far" in Hamming distance from any codeword, while only probing a small fraction of the input. Much stronger upper bounds are known for locally testable codes than for LDCs, and in particular, there exists $O(1)$-local LTCs with blocklength $n = k \cdot \text{polylog}(k)$ [GS06] (see also [Mei09, Vid13]). It is also known that LDCs do *not* imply locally testable codes and vice versa [KV10].

Locally testable codes can be viewed as a special case of *property testing* (see recent book [Gol17] and references therein), which deals with algorithms that distinguish whether an input belongs to a set $S$ or is "far" from any input in $S$. As is the case with locally testable codes, LDCs and property testing are very distinct notions. In particular, whereas a local decoder is a *local computation* algorithm that operates under the guarantee that the input is *close to a codeword*, a property tester is an *approximate decision* algorithm that distinguishes between exact membership in a set, and being *far* from the set. Interestingly, despite these fundamental differences, we are still able to rely on techniques from [FLV15] that were used in the context of property testing (see Section 2.4).

### 1.3 Organisation

The rest of the paper is organised as follows. In Section 2 we provide a high-level overview of our techniques. In Section 3 we cover the necessary preliminaries. Finally, in Section 4 we prove Theorem 1.

## 2 Techniques

In this section, we provide an overview of the proof of the lower bound in Theorem 1. We begin in Section 2.1 by articulating the challenge in proving a lower bound on relaxed LDCs and discuss why current techniques for non-relaxed LDC lower bounds are inherently incompatible with the setting of relaxed LDCs.

In Section 2.2, we present our high-level strategy for obtaining the lower bound, which is centred around using the relaxed local decoder to obtain a "global decoder"; that is, a probabilistic algorithm that decodes the *entire* message of a *perfectly valid* codeword. In Section 2.3, we discuss a naive

attempt towards constructing such a global decoder, and articulate two main technical challenges that arise.

In Section 2.4, we address the first challenge by arguing that the local views of relaxed local decoders can be assumed, without loss of generality, to satisfy a structure that can be thought of as a relaxation of combinatorial sunflowers. In Section 2.5, we address the remaining challenge and present the construction of our global decoder. Finally, in Section 2.6, we discuss the analysis of the global decoder and how it implies the desired lower bound.

## 2.1 The challenge

As we mentioned in Section 1.2, the coding theory literature has a large body of works that prove lower bounds on *non-relaxed* LDCs. It is tempting to try and apply the methodology used in these works to our setting of *relaxed* LDCs.

The caveat, however, is that essentially all LDC lower bound techniques in the literature rely on the *smoothness* property of LDCs (cf. [KT00, Theorem 1]). Loosely speaking, a decoder is said to be smooth if the distribution of queries that it makes is well-spread; that is, no coordinate is being queried with high probability by the decoder. The smoothness of LDCs provides structural insight regarding local decoders, which lie at the heart of these techniques.

In stark contrast, relaxed LDCs are *not* necessarily smooth. In fact, all known constructions of non-trivial relaxed LDCs (i.e., which achieve parameters that are better than known for non-relaxed LDCs) are highly non-smooth, in the following sense: for each message index $i \in [k]$, a significant fraction of the queries that the relaxed local decoder makes are concentrated on a small number of coordinates.

This lack of smoothness is inherent to *relaxed* LDCs, as it is known that if a relaxed LDC is smooth, then it implies a non-relaxed LDC with similar parameters [BDG17]. Thus, if it was possible to make state-of-the-art relaxed LDCs smooth, it would have led to a major breakthrough, as this would imply non-relaxed LDCs with polynomial length and $O(1)$-queries.

As observed in [BGH$^+$04], relaxed LDCs can be made to satisfy a weaker condition, known as *average smoothness*, which states that the decoder makes nearly uniform queries on *average*, taken over all indices $i \in [k]$ to be decoded (however, for any particular $i \in [k]$, the queries of $D$ given decoding index $i$ may be highly concentrated). Unfortunately, the average smoothness condition is a much weaker requirement than smoothness (e.g., see discussion in [BGH$^+$04, Section 4.2.1]), and it is highly unclear whether it can be used to imply relaxed LDC lower bounds.

Instead, to show a lower bound on relaxed LDCs we use a new methodology that does *not* rely on smoothness at all to argue about the structure of the relaxed local decoder. The approach that we take, which we discuss next in Section 2.2, strongly relies on an observation that the structure of the local views that relaxed decoders make can be essentially captured by a relaxation of the notion of sunflowers, to which we refer as *daisies* and discuss in Section 2.4. We stress that despite some points of similarity, our techniques are fundamentally different than those in [KT00] (see discussion at the end of Section 2.6).

## 2.2 High-level approach

Recall that our goal is to show that every $O(1)$-local relaxed LDC $C\colon \{0,1\}^k \to \{0,1\}^n$ with decoding radius $\delta = O(1)$ satisfies $n = \Omega\left(k^{1+\alpha}\right)$, for some constant $\alpha > 0$ that depends on the locality parameter and decoding radius.

Let $C$ be such $\ell$-local relaxed LDC for $\ell = O(1)$, and let $D$ be its corresponding relaxed local decoder. For clarity of exposition, throughout the techniques section we make the simplifying assumptions that $D$ has the following properties:

1. *non-adaptive queries:* each query is made independently of the answers to previous queries,
2. *reduced error probability:* the decoder errs with probability $O(1/\ell^2)$,
3. *logarithmic randomness complexity:* the decoder uses $\log(n) + O(1)$ bits to generate its queries.

In the actual proof, we obtain these properties by adapting standard transformations to our setting, at the cost of deterioration in part of the parameters;[1] see Section 4.1 for details.

Our strategy for proving Theorem 1 is to rely on the relaxed local decoder $D$ to construct a *sample-based*, *global* decoder $G$ for the code $C$. By sample based, we mean that the decoder $G$ queries each coordinate independently with a certain probability $p$. By global, we mean that $G$ decodes the *entire* message of a *perfectly valid* codeword. We stress that for our argument, it suffices for the global decoder to only work under the promise that the input codeword is not corrupted.

Our goal is to show that the global decoder $G$ will successfully decode the entire message, when the sampling parameter is set as $p = 1/n^{1/2\ell^2}$; we discuss this choice of $p$ in Section 2.5. Note that in this case, with high probability $G$ only makes $O(n^{1-1/2\ell^2})$ queries to the input (and so, if it exceeds the desired query complexity, it can simply reject). However, since it is information theoretically impossible to recover a $k$-bit message via $o(k)$ queries, this will imply that $n = \Omega\left(k^{1+\frac{1}{2\ell^2-1}}\right)$, which yields the desired lower bound. See Section 4.5 for a precise argument.

We remark that the global decoder can be viewed as showing that any $\ell$-local relaxed LDC is recoverable with high probability from the binary erasure channel (BEC) with erasure probability $1 - 1/n^{1/2\ell^2}$.

Thus, we are left with the task of showing that the set of queries that $G$ makes with parameter $p = 1/n^{1/2\ell^2}$ can suffice for simultaneously emulating $k$ invocations of the decoder $D$ with respect to each decoding index $i \in [k]$. To this end, we shall first need to make a simple, yet important observation regarding relaxed local decoders, which we discuss next.

### 2.3 First step towards a global decoder

Recall that we assumed the relaxed local decoder $D$ is a non-adaptive algorithm with logarithmic randomness complexity, which gets query access to a string $w \in \{0,1\}^n$. Thus, we can represent $D$ as a collection of distributions $\{\mu_i\}_{i\in[k]}$ over subsets of $[n]$ of size $\ell$, and functions $\{f_i \colon \{0,1\}^\ell \to \{0,1,\perp\}\}_{i\in[k]}$ as follows.

For every $i \in [k]$, the distribution $\mu_i$ corresponds to the choice of local view of the relaxed decoder $D(i)$ (i.e., $D$ on decoding index $i$). The function $f_i$ is the predicate according to which $D(i)$ decides whether to decode 0, 1, or reject (output $\perp$) given a local view $w|_I$, where $I$ is drawn from $\mu_i$.[2] Note that since we assumed that $D(i)$ has logarithmic randomness complexity (i.e., $\log(n) + O(1)$, in the blocklength $n$), we have that $\mu_i$ is supported on a linear number of local views (i.e., sets of size at most $\ell$); we use this in Section 2.4, where we apply a combinatorial lemma on the support of $\mu_i$.

[1] The loss in parameters due to the last two transformations is minor. The adaptive to non-adaptive transformation, on the other hand, increases the (constant) query complexity by an exponential factor.

[2] In fact, the decoder may rule according to a predicate that depends on the query set, but we ignore this subtlety in the high-level overview.

Naively, we would have liked our global decoder $G$, which queries each location with probability $p = 1/n^{1/2\ell^2}$, to emulate an invocation of the relaxed local decoder $D(i)$ by obtaining a local view of $w$ restricted to $I \sim \mu_i$. Indeed, if the distribution $\mu_i$ is "well spread", the probability of obtaining a local view of $D(i)$ is high. Suppose, for instance, that all of the local views are pairwise disjoint. In this case, the probability of $G$ obtaining any particular local view is at least $p^\ell = 1/n^{1/2\ell}$, and since there are $\Theta(n)$ such local views, we can expect the global decoder to obtain $\Theta(n^{1-\frac{1}{2\ell}})$ local views.

Unfortunately, if $\mu_i$ is concentrated on a relatively small number of coordinates (as is the case with all non-trivial relaxed local decoders), it is highly unlikely that the global decoder $G$ will obtain a local view of $D(i)$. For example, if $D(i)$ queries the first coordinate of $w$ with probability 1, then we can obtain a local view of $D(i)$ with probability at most $p$, which is negligible.

Even worse, to decode different bits, the relaxed local decoder $D$ may concentrate its queries on different locations; e.g., it could be the case that for every $i \in [k]$, the decoder $D$ would query, say, location $i$ with probability 1. And so, there could be a large number of coordinates that are heavily queried by $D$.

At this point, the approach may appear hopeless. However, this is exactly where the relaxed decoding condition of relaxed LDCs kicks in. Recall that the relaxed local decoder $D$ does not err (i.e., outputs the wrong value) with high probability, as long as the codeword is not too corrupted. Thus, even if we arbitrarily guess the values of highly queried coordinates that the global decoder $G$ failed to achieve, we could still emulate an invocation of $D$ on a slightly corrupted codeword.

For example, suppose that all of the local views of the relaxed local decoder $D$ contain, say, the first coordinate, but are otherwise disjoint. Then, by the discussion above, with high probability the global decoder $G$ will obtain many partial local views, which only lack the value of the first coordinate. We can then hope to rely on the ability of the relaxed local decoder $D$ to tolerate errors, and arbitrarily fill in the value of the missing coordinate.

Namely, we could consider both possible values of the first coordinate, and observe the following. For the right "guess" of the value of the first coordinate, *all* local views will lead $D$ to decode correctly, whereas for the incorrect guess, the *majority* of local views will lead $D$ to either the correct value or $\perp$, preventing a consensus on the wrong value. (See detailed discussion of this approach in Section 2.5.)

However, there are two main challenges that arise when attempting to implement the foregoing strategy in the general case; namely:

1. The combinatorial structure of the local views of a relaxed local decoder may be complex and involve many intersections; and

2. Unlike in (non-relaxed) LDCs, a *relaxed* decoder may output a reject symbol $\perp$ with high probability (possibly with probability 1), even if only one bit of the codeword is corrupted.

We address the first challenge in Section 2.4, in which we make a crucial observation about the combinatorial structure of relaxed local decoders, and address the second challenge in Section 2.5, where we describe our construction of the global decoder and its analysis.

### 2.4 Relaxed sunflowers

For the next discussion, fix $i \in [k]$, and denote by $\{ L_m \}_m$ the set of all local views the relaxed local decoder $D$ might query on explicit input $i$. Recall that $D(i)$ queries $L_m$ with probability $\mu_i(L_m)$.

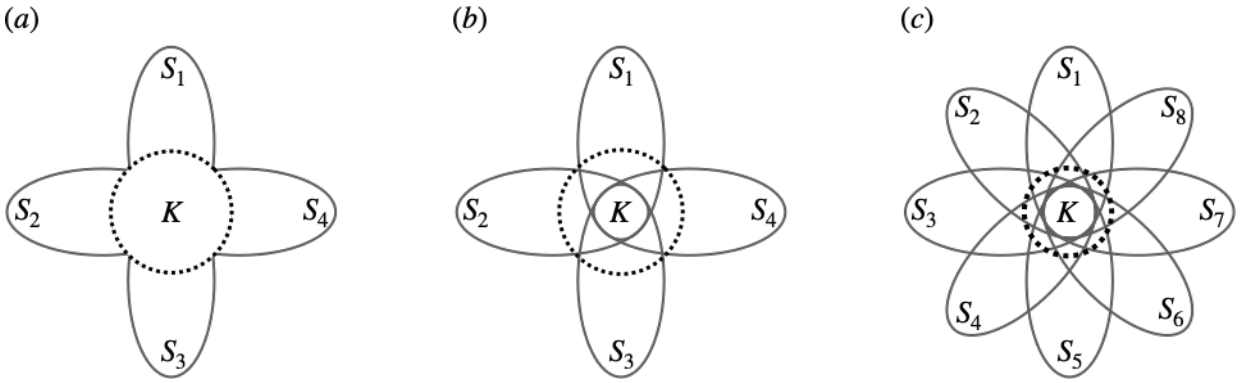


Figure 1: (a) *sunflower*: all sets $\{ S_m \}_m$ intersect on the kernel $K$ and are otherwise pairwise disjoint; (b) *simple daisy (a.k.a., 1-daisy)*: outside of the kernel $K$, all sets $\{ S_m \}_m$ are pairwise disjoint; and (c) *t-daisy*: outside of the kernel $K$, each point is covered by at most $t$ sets in $\{ S_m \}_m$. In all figures, the dashed circle represents the kernel $K$.

In Section 2.3, we observed that if the sets $\{ L_m \}_m$ intersect on a single coordinate, i.e., $\cap_m L_m = j$ for some $j \in [n]$, and are otherwise pairwise disjoint, then the global decoder $G$ queries many partial sets $L_m \setminus \{ j \}$ with high probability, where not knowing the value at coordinate $j$ still leaves us with an input within the decoding radius.

More generally, since relaxed LDCs can tolerate a large (constant) fraction of errors, the foregoing argument can be extended to combinatorial *sunflowers*; that is, a collection $\mathcal{S} \coloneqq \{ S_m \}_m$ of subsets of $[n]$ for which there exist a "kernel" $K \subseteq [n]$ such that: (1) $\cap_m S_m = K$, and (2) the "petals" $\mathcal{P} = \{ S_m \setminus K \}$ are pairwise disjoint (see Fig. 1(a)), where the kernel is small enough such that by changing the values we assign to it we will still remain within the decoding radius.

Of course, there is no guarantee that the local views of a relaxed local decoder will be *sunflowers*. While we could use *sunflower lemmas* to extract sunflowers out of an arbitrary collection of subsets, we stress that the size of such sunflowers is very small (in particular, *sub-linear*). Hence, we cannot simply restrict our attention to a subset of the local views that is a sunflower, as this would not preserve the soundness of the relaxed local decoder.

Nevertheless, we can use the relaxed decoding condition of $D$ even if our local views satisfy a less rigid structure than that of a sunflower. Specifically, for our argument to go through, we need the local views $\{ S_m \}_m$ to only be such that outside of an arbitrarily-structured set $K$ of small density, each point is contained in a small (sublinear) number of petals.

Fortunately, Fischer et al. [FLV15] encountered a similar combinatorial structure in the setting of *property testing*, which led them to define a couple of generalisations of sunflowers, to which they refer as "pompoms" and "constellations", and prove combinatorial lemmas regarding them. Following [FLV15], we consider a closely related relaxation of sunflowers, which we call *daisies*.

Loosely speaking, a *daisy* is a sunflower in which the kernel is not necessarily the intersection of all petals, but rather a small subset such that every element outside the kernel is contained in a small number of petals.

More precisely, a collection $\mathcal{S} \coloneqq \{ S_m \}_m$ of subsets of $[n]$ is a $t$-daisy with respect to a *kernel* $K \subseteq [n]$ if for every element $j \in [n] \setminus K$, there are at most $t$ subsets $S \in \mathcal{S}$ such that $j$ is contained in the *petal* $S \setminus K$. We will refer to the special case of a 1-daisy, wherein the petals are disjoint, as a *simple daisy*. (See Fig. 1 (b) and (c).)

Using techniques developed in [FLV15], we show a *daisy lemma* (conceptually resembling a sunflower lemma) that extracts a $t$-daisy of large (constant) density, with a small kernel (i.e., such that changing its values would keep us within the decoding radius), and where $t$ is sufficiently small for obtaining petals, with high probability, using the sampling that the global decoder $G$ performs. See Lemma 4.6 for a precise statement. We remark that the daisy lemma applies to any collection of (possibly weighted) subsets, and does not rely on the fact that our collection arises from a relaxed LDC.

In more detail, the daisy lemma extracts a $t$-daisy from the local views of each $D(i)$, which satisfies the following conditions:

- the density of the daisy is at least $1/\ell$;
- the size of the kernel $K_i$ is roughly $n^{1-\frac{s}{\ell}}$, where $s \in [\ell]$ bounds the maximal size of the petals;
- the number of sets that cover each point outside of the kernel is $t = O(n^{\frac{s-1}{\ell}})$.

Next, we shall use the foregoing daisy lemma to construct the global decoder.

### 2.5 The global decoder

We are finally ready to describe our construction of the global decoder $G$ for the code $C\colon \{0,1\}^k \to \{0,1\}^n$, using the relaxed local decoder $D$.

The global decoder $G$ is given query access to a string $w \in \{0,1\}^n$, promised to satisfy $w = C(x)$, for $x \in \{0,1\}^k$. Its goal is to fully decode $x$. To this end, $G$ starts by querying each coordinate $w_j$, for $j \in [n]$, independently with probability $p = 1/n^{1/2\ell^2}$, and tries to obtain local views of the relaxed local decoder $D(i)$ for each location $i \in [k]$, while *reusing* the same samples.

Recall that the structure of the local views of $D$ does not guarantee that any local view will be fully queried in the sampling stage of the global decoder $G$. However, we can invoke the daisy lemma that we discussed in Section 2.4 to extract from $\mathrm{supp}(\mu_i)$ a sub-collection of sets, of total density at least $1/\ell$, which is a daisy $\mathcal{S}_i$ with kernel $K_i$, for each $i \in [k]$. Recall that the soundness error of $D$ is $1/\ell^2$, and so the density of the daisy is significantly larger than the soundness error.

Now, intuitively, since the petals of each daisy (i.e., sets $S \setminus K$ for $S \in \mathcal{S}_i$) are such that each point is contained in a small (sublinear) number of petals, we can expect the global decoder $G$ to fully query a large number of petals of each daisy in $\{\mathcal{S}_i\}_{i\in[k]}$. (See more on this at the end of Section 2.5.) However, it may happen that *none* of the kernels $K_1, \ldots, K_k$ is queried at all (let alone fully queried) by $G$.

Thus, the main challenge is to recover the value of each $x_i$ using partial local views of $D(i)$ that do not include the value of the input $w = C(x)$ on the kernel $K_i$. For this, we shall need to rely on the properties of the relaxed local decoder, and the fact that the size of each $K_i$ is small enough such that by changing it we remain within the decoding radius of the relaxed LDC $C$.

The idea is to let the global decoder $G$ consider all possible assignments to the kernel, use each such assignment to complete the queried petals into full local views of the relaxed local decoder $D$, and rely on the properties of relaxed LDCs to identify a kernel assignment that corresponds to the decoding the correct value.

More precisely, for each $i \in [k]$, the global decoder $G$ enumerates over all possible assignments $\kappa \in \{0,1\}^{|K_i|}$ to the kernel, and for every fully queried petal $P$ of $\mathcal{S}_i$, considers the output of $D(i)$ on each local view $S \in \mathcal{S}_i$ that consists of $w$ restricted to the petal $P$, with the value of $\kappa$ on the kernel $K_i$.

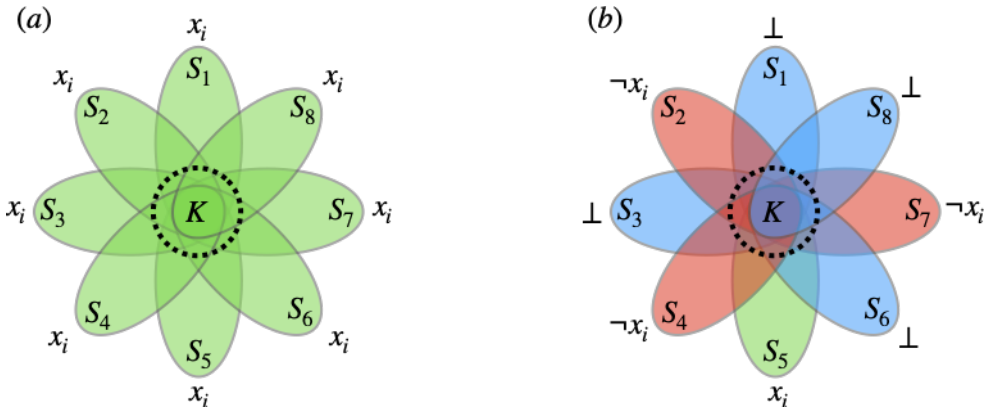


Figure 2: (a) corresponds to a correct "guess" of the kernel assignment, and (b) to a wrong one. Green sets correspond to local views that make the relaxed local decoder output the *correct* value $x_i$. Red sets correspond to local views that lead to outputting the *wrong* value $\neg x_i$. Blue sets correspond to local views that lead to reject, i.e., output $\perp$.

Now, it is crucial to make the observation that by the completeness and relaxed decoding conditions of relaxed LDCs, no kernel assignment will give rise to a majority of petals that leads to decoding the wrong value, whereas there exists a least one kernel assignment that will make all petals lead to decode the correct value. More accurately, since the density of the daisy $\mathcal{S}_i$ is at least $1/\ell$, and the soundness error is at most $1/\ell^2$, we have that:

1. Since $w$ is guaranteed to be a valid codeword $C(x)$, then for the correct assignment to the kernel $K_i$ (i.e., $\kappa \in \{0,1\}^{|K_i|}$ such that $\kappa = w|_{K_i}$), it holds that all local views $S \in \mathcal{S}_i$ would have made the decoder $D(i)$ output the correct value $x_i$ (see Fig. 2(a)); and,

2. Since changing the value of the kernel $K_i$ still leaves us within the decoding radius, then for any kernel assignment $\kappa \in \{0,1\}^{|K_i|}$, the majority of local views $S \in \mathcal{S}_i$ would have *not* made the decoder $D(i)$ output the wrong value $\neg x_i$ (see Fig. 2(b)).

The foregoing discussion naturally suggests that $G$ will decode each $x_i$ as follows: if there exists a kernel assignment that completes all fully-queried petals to local views that are in consensus on a single value $b$, and no kernel assignment leads to a consensus on $\neg b$, then output $b$. We will show that this indeed happens with $b = x_i$.

To see that, note that the first item above guarantees that there exists at least one kernel assignment for each $K_i$, which makes all petals lead to decoding the correct value $x_i$. The second item guarantees that, with high probability over petals of $\mathcal{S}_i$ that $G$ fully queried, no kernel assignment will lead the majority of completions of fully-queried petals to local views that are consistent with the wrong value $\neg x_i$. Thus the global decoder can enumerate over all kernel assignments, and decode according to the kernel assignment that leads to a consensus.

We stress that the global decoder $G$ *cannot* just guess an arbitrary value of the kernel (which will still leave us within the decoding radius). This is because it could be the case that for some kernel assignments, the majority of petals will lead to $\perp$, and petals that lead to decoding the wrong value $\neg x_i$ will be actually *more common* than petals that lead to the correct value $x_i$. However, the key point is that no *majority* of petals will lead to decoding the wrong value, whereas there exists a kernel assignment that will lead all petals to decode the correct value.

### 2.6 Analysis of the global decoder

It remains to argue that with high probability, for every $i \in [k]$ the global decoder $G$, described in Section 2.5, will successfully obtain fully-queried petals that will lead it to correctly decode $x_i$; that is, a set of petals such that: (1) no kernel assignment will give rise to a majority of petals that lead to decode the wrong value, and (2) there exists at least one kernel assignment that will make all petals lead to decode the correct value.

Observe that it suffices to show that for every $i \in [k]$ and kernel assignment $\kappa \in \{0,1\}^{|K_i|}$, the global decoder $G$ only needs to obtain a single petal that leads to either outputting the correct value $x_i$ or the reject symbol $\perp$, given the kernel assignment $\kappa$; we shall refer to such petals as "good". To see this, note that $G$ only accepts if all petals it queried are in consensus regarding the decoding value, and so, as long there is at least one petal that corresponds to $x_i$ or $\perp$, the global decoder $G$ will not output the wrong value $\neg x_i$. On the other hand, we know that there exists a kernel assignment for which all petals lead to output the correct value $x_i$ (and neither $\perp$ nor $\neg x_i$), and so, as long as $G$ obtains one petal, it will output $x_i$.

To show that for every $i \in [k]$ and kernel assignment $\kappa \in \{0,1\}^{|K_i|}$, the global decoder $G$ obtains a good petal as above, we first remove all petals that lead to decoding the wrong value. Recall that the density of the daisy $\mathcal{S}_i$ we obtained from the daisy lemma is at least $1/\ell$, and the soundness error of the relaxed local decoder $D$ is at most $1/\ell^2$. Thus the total density of the good petals is at least $1/\ell - 1/\ell^2$.[3]

Next, we show that the density of the good petals implies that there are many of them. To this end, observe that since the density of the good local views is larger than the soundness error, then the fractional size of the set of all elements covered by the good petals must be larger than the (constant) decoding radius. This is because otherwise, replacing these elements with the values of a codeword that disagrees with $x_i$ will leave $w$ within the decoding radius, and thus break the soundness condition. Hence, the good petals cover a linear amount of coordinates, and since each petal is of constant size, we have a linear number of good petals.

Since there are many good petals in the daisy $\mathcal{S}_i$, we can apply a lemma, which we call the *simple daisy lemma* (see Lemma 4.7), that extracts a simple daisy (i.e., a 1-daisy in which the petals are pairwise disjoint) from the set of all good local views in our $t$-daisy (where recall that $t = O(n^{\frac{s-1}{\ell}})$ and $s$ bounds the maximal size of each of the petals). The resulting simple daisy has the same kernel $K_i$ of size roughly $n^{1-\frac{s}{\ell}}$, and number of petals, each of size at most $s$, that is larger than the size of kernel $K_i$ by a multiplicative factor of at least $n^{1/\ell}$.

The petals of a simple daisy are *disjoint*, and so, observe that the probability of querying all (at most $s$) elements of any petal during the sampling step is at least $p^s = 1/n^{s/2\ell^2} \geq 1/n^{1/2\ell}$. Since our simple daisy contains $d \coloneqq \Omega(n^{\frac{1}{\ell}} \cdot n^{1-\frac{s}{\ell}})$ pairwise disjoint petals, the probability that no good petal was queried is bounded by

$$(1-p^s)^d \leq \left(1 - \frac{1}{n^{\frac{1}{2\ell}}}\right)^{\Omega(n^{\frac{1}{\ell}} \cdot n^{1-\frac{s}{\ell}})} \leq \frac{2^{-\Omega(|K_i|)}}{10k} \ .$$

Thus, for any $i \in [k]$, taking a union bound over all kernel assignments $\kappa \in \{0,1\}^{|K_i|}$, with probability at least $1 - 1/(10k)$, for every kernel assignment we find a least one good petal, which by the discussion above implies that the global decoder $G$ successfully decodes the correct value $x_i$.

[3] In fact, it suffices have soundness error, say, $1/(10\ell)$. However, reducing the error to $1/\ell^2$ has negligible cost and it makes calculations slightly cleaner.

Finally, taking another union bound over all decoding indices $i \in [k]$, we obtain that the foregoing holds for all $i \in [k]$ *simultaneously* with probability at least 9/10.

As discussed in Section 2.2, since it is information theoretically impossible to recover a $k$-bit message via $o(k)$ queries, and since the global decoder $G$ decoders $k$ message bits via $O(n^{1-1/2\ell^2})$ queries to the input, we deduce that $n = \Omega\left(k^{1+\frac{1}{2\ell^2-1}}\right)$, which yields the desired lower bound.

**Comparison to the techniques in [KT00].** While on a high level, there are some similarities between our techniques and the ones in [KT00], we wish to highlight that our techniques are fundamentally different.

For starters, as we alluded to before, the approach in [KT00] relies on the equivalence of LDCs and smooth codes, and uses the smoothness in an essential way. Relaxed LDC, on the other hand, can be very far from smooth; in fact, the strongest constructions of relaxed LDC admit $k$ different query distributions (one per decoding index), where in each such distribution, different points are being queried with probability 1. Thus, relaxed LDCs cannot be lower bounded by identifying a small set of locations (a "kernel") such that the decoder is uniform outside of this set. Indeed, note that for any coordinate $i \in [k]$, we choose a *different* daisy.

Moreover, a simple daisy (i.e., where the petals are pairwise disjoint) cannot be large enough to characterise a relaxed decoder, *even for a single coordinate*. This is the reason why we cannot combine the daisy lemma with the simple-daisy lemma and restrict our attention to the latter. We only use simple daisies for analysing the probability of querying a good petal of a $t$-daisy, conditioned on fixing a particular kernel assignment; and so, for each kernel assignment to a $t$-daisy, we extract *a different simple daisy*. Indeed, our algorithm operates by checking assignments to $k$ different, large $t$-daisies, for $t = n^{\beta}$, and so for each of these $t$-daisies, the *queries outside of the kernel are also very far from being smooth*.

We believe that our new techniques may be also useful for locally testable codes, and hopefully (albeit within a different framework) the combinatorial tool of daisies could lead to progress on non-relaxed LDCs as well.

# 3 Preliminaries

We begin with standard notation:

- We denote the *absolute distance*, over alphabet $\Sigma$, between two strings $x \in \Sigma^n$ and $y \in \Sigma^n$ by $\bar{\Delta}(x, y) \coloneqq |\{x_i \neq y_i : \ i \in [n]\}|$ and their *relative distance* by $\Delta(x, y) \coloneqq \frac{\bar{\Delta}(x,y)}{n}$. If $\Delta(x, y) \leq \varepsilon$, we say that $x$ is $\varepsilon$-*close* to $y$, and otherwise we say that $x$ is $\varepsilon$-*far* from $y$. Similarly, we denote the *absolute distance* of $x$ from a non-empty set $S \subseteq \Sigma^n$ by $\bar{\Delta}(x, S) \coloneqq \min_{y \in S} \bar{\Delta}(x, y)$ and the *relative distance* of $x$ from $S$ by $\Delta(x, S) \coloneqq \min_{y \in S} \Delta(x, y)$. If $\Delta(x, S) \leq \varepsilon$, we say that $x$ is $\varepsilon$-*close* to $S$, and otherwise we say that $x$ is $\varepsilon$-*far* from $S$. We denote the projection of $x \in \Sigma^n$ on $I \subseteq [n]$ by $x|_I$.

- We denote by $A^x(y)$ the output of algorithm $A$ given direct access to input $y$ and oracle access to string $x$. Given two interactive machines $A$ and $B$, we denote by $(A^x, B(y))(z)$ the output of $A$ when interacting with $B$, where $A$ (respectively, $B$) is given oracle access to $x$ (respectively, direct access to $y$) and both parties have direct access to $z$.

- Throughout this work, probabilistic expressions that involve a randomised algorithm $A$ are taken over the inner randomness of $A$ (e.g., when we write $\Pr[A^x(y) = z]$, the probability is taken over the coin-tosses of $A$).

### 3.1 Coding theory

Let $k < n$ be positive integers and let $\Gamma, \Sigma$ be alphabets. A *code* $C\colon \Gamma^k \to \Sigma^n$ is an *injective* mapping from messages of length $k$ (over the alphabet $\Gamma$) to codewords of length $n$ (over the alphabet $\Sigma$). Typically it will be the case that $\Gamma = \Sigma$, in which case we simply say that the code is over the alphabet $\Sigma$. We denote by $n$ the *blocklength* of the code (which we think of as a function of $k$) and by $k/n$ the *rate* of the code. The *relative distance* of the code is the minimum, over all distinct messages $x, y \in \Gamma^k$, of $\Delta(C(x), C(y))$. We shall sometimes slightly abuse notation and use $C$ to denote the set of all of its codewords $\{C(x)\}_{x\in\Gamma^k} \subset \Sigma^n$.

### 3.2 Locally decodable codes

First, we define the notion of (non-relaxed) locally decodable codes.

**Definition 3.1** (Locally Decodable Codes (LDCs))**.** *Let $C \subseteq \Sigma^n$ be a code with relative distance $\delta_C$. We say that $\mathcal{C}$ is a* locally decodable code *if there exists a constant* decoding radius $\delta < \delta_C/2$ *and a polynomial time algorithm $D$ that gets oracle access to a string $w \in \Sigma^n$ and explicit input $i \in [k]$, such that*

1. *(Perfect) Completeness: For any $i \in [k]$ and $w = C(x)$, where $x \in \Sigma^k$, it holds that $D^w(i) = x_i$.*
2. *Decoding: For any $i \in [k]$ and any $w \in \Sigma^n$ that is $\delta$-close to a (unique) codeword $C(x)$,*
$$\Pr[D^w(i) = x_i] \geq 2/3.$$

*The* query complexity *of $D$ is the maximal number of queries that $D$ makes for any input $i$ and $w$.*

Note that the constant $2/3$ can be amplified as usual by repeating the process multiple times and outputting the majority symbol.

Relaxed locally decodable codes [BGH+04] are defined as follows.

**Definition 3.2** (RLDC)**.** *A code $C\colon \Sigma^k \to \Sigma^n$ is an $\ell$-local relaxed LDC (RLDC) eith success rate $\rho$ if there exists a constant $\delta \in (0, \delta_C/2)$ and a randomised algorithm $D$, known as a* relaxed decoder*, that on explicit input $i \in [k]$ makes $\ell$ queries to an oracle $w$ and satisfies the following conditions.*

1. *(Perfect) Completeness: For any $i \in [k]$ and $w = C(x)$, where $x \in \Sigma^k$, it holds that $D^w(i) = x_i$.*
2. *Relaxed Decoding: For any $i \in [k]$ and any $w \in \Sigma^n$ that is $\delta$-close to a (unique) codeword $C(x)$,*
$$\Pr[D^w(i) \in \{x_i, \perp\}] \geq 2/3.$$
3. *Success Rate: There exists a constant $\rho > 0$ such that for any $w \in \{0,1\}^n$ that is $\delta$-close to a codeword $C(x)$, there exists a set $I_w \subseteq [k]$ of size at least $\rho k$ such that for every $i \in I_w$,*
$$\Pr[D^w(i) = x_i] \geq 2/3 \ .$$

The *randomness complexity* of a relaxed decoder is the maximal number of random coin tosses it requires to select its local view (i.e., the set of queries it makes), where the maximum is taken over the index $i \in [k]$ with respect to which it is invoked.

# 4 Proof of Theorem 1

We begin by restating Theorem 1 more accurately.

**Theorem 4.1.** *For any $\ell \in \mathbb{N}$ and $\delta \in (0,1)$, there exists a constant $\alpha = \alpha(\ell, \delta) > 0$ such that every $\ell$-local relaxed LDC $C\colon \{0,1\}^k \to \{0,1\}^n$ with decoding radius $\delta$ satisfies*

$$n = \Omega\left(k^{1+\frac{1}{2^{2\ell}\cdot\log(\ell)^2-1}}\right) \ .$$

Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be an $\ell$-local relaxed LDC with decoding radius $\delta = \Omega(1)$ and $\ell = O(1)$; denote its decoder by $D$. We prove that the blocklength of $C$ is as in the conclusion of Theorem 4.1. To this end, we begin in Section 4.1 by preprocessing the relaxed local decoder $D$ and endowing it with properties that make it amenable to our techniques. Then, in Section 4.2 we present two combinatorial lemmas that will play a key role in our analysis, allowing us to structurally argue about relaxed decoders via subsets of their local views.

Next, in Section 4.3 we implement the strategy presented in Section 2, by using the relaxed local decoder to obtain a construction of a global decoder $G$ that receives a *valid* codeword and decodes its entire message using query complexity that is sublinear in the blocklength $n$. We analyse the global decoder in Section 4.4, and finally, in Section 4.5 we derive the lower bound in Theorem 1 from the analysis of the global decoder.

## 4.1 Preprocessing

Our argument begins by endowing the decoder $D$ with three properties that will facilitate the analysis of our lower bound, namely: (1) non-adaptive queries, (2) reduced error probability, and (3) logarithmic randomness complexity.

All three steps of the preprocessing step follow from straightforward adaptation of standard techniques to the setting of relaxed LDCs. We defer their proofs to Appendix A. We begin by transforming the decoder $D$ into a non-adaptive algorithm via a standard transformation. We note that this transformation increases the query complexity, while preserving or improving the rest of the parameters of $D$. This will later allow us to represent the behaviour of a relaxed decoder by the distribution over its local views.

**Claim 4.2** (Non-adaptive queries)**.** *If there exists an (adaptive) $\ell$-local relaxed decoder for a code $C\colon \{0,1\}^k \to \{0,1\}^n$, then $C$ also has a* non-adaptive *$2^\ell$-local relaxed decoder with the same decoding radius.*

See Appendix A.2 for the proof of Claim 4.2. Denote by $D_1$ the non-adaptive relaxed decoder obtained by applying Claim 4.2 to the (adaptive) relaxed decoder $D$.

Next, we amplify the soundness of the decoder $D_1$, as later we shall need to invoke it multiple times and tolerate a union bound over all invocations. To this end, we use the following simple claim.

**Claim 4.3** (Amplification)**.** *If there exists a non-adaptive $\ell$-local relaxed decoder for the code $C$, which errs with probability at most $1/3$, then $C$ also has an amplified non-adaptive $O(\ell \cdot \log(1/\varepsilon))$-local relaxed decoder that errs with probability at most $\epsilon$; furthermore, the amplified relaxed decoder preserves the perfect completeness condition.*

See Appendix A.3 for the proof of Claim 4.3. We remark that while the foregoing proof is trivial, there is a subtlety that is unique to the setting of *relaxed* LDC. Namely, The transformation in Claim 4.3 hampers the success rate condition of a relaxed LDC,[4] however, we stress that our argument does *not* rely on the aforementioned condition. Denote by $D_2$ the amplified relaxed decoder obtained by applying Claim 4.3 to the relaxed decoder $D_1$ with respect to soundness error $\varepsilon = 1/\ell^2$.

Finally, we generalise a lemma due to Goldreich and Sheffet [GS10], which reduces the randomness complexity of query algorithms, to the setting of relaxed LDCs. This will later allow us to invoke a relaxed sunflower lemma (see Section 4.2).

**Claim 4.4** (Randomness reduction)**.** *If there exists a non-adaptive, $\ell$-local relaxed decoder, where $\ell = O(1)$, for a* binary *code $C$ with constant error probability $\varepsilon$, then $C$ also has an $O(\ell)$-local relaxed local decoder with the same parameters, except with randomness complexity at most* $\log(n) + O(1)$.

See Appendix A.4 for the proof of Claim 4.4. Denote by $D'$ the relaxed decoder obtained by applying the randomness reduction in Claim 4.4 to the relaxed decoder $D_2$.

We conclude this subsection by observing that $D'$ is a *non-adaptive* $\ell'$-local relaxed decoder with soundness error $\varepsilon'$ and randomness complexity $r'$, where $\ell' = O(2^\ell \cdot \log(\ell)) = O(1)$, $\varepsilon' = 1/\ell'^2$, and $r' = \log(n) + O(1)$.

## 4.2 Relaxed sunflower lemmas

As discussed in the technical overview, we shall view the non-adaptive $\ell'$-local relaxed decoder $D'$ obtained in Section 4.1 as a set of distributions $\{ \mu_i \}_{i \in [k]}$, where each $\mu_i$ is the distribution over subsets of $[n]$ of size $\ell'$, which correspond to the local views of the relaxed decoder $D'$ on decoding index $i \in [k]$.

To argue about the probability of obtaining local views of a (non-adaptive) relaxed local decoder, we shall consider a structured subset of the local views, which satisfies a relaxed form of a combinatorial sunflower, to which we refer to as a *daisy*. Loosely speaking, a $t$-daisy is a sunflower in which the kernel is not the intersection of all petals, but rather a subset such that every element outside the kernel is contained in at most $t$ petals. A formal definition follows.

**Definition 4.5** (daisy)**.** *Suppose $U$ is a universe set and $\mathcal{S}$ is a collection of subsets of $U$. The collection $\mathcal{S}$ is an* $t$-daisy *with a* kernel *$K \subseteq U$, if for every $u \in U \setminus K$ there are at most $t$ subsets $S \in \mathcal{S}$ such that $u$ is contained in the* petal *$S \setminus K$.*

We will refer to the special case of a 1-daisy, wherein the petals are disjoint, as a *simple daisy*.

We remark that the notion of a daisy is a generalisation of a sunflower, which provides a unified view of the notions of "pompoms" and "constellations", defined in [FLV15], without insisting on petals of equal size.

In the following it will be convenient to define the *degree* of an element $u \in U$ in a collection $\mathcal{S} \subseteq 2^U$ by $\deg_{\mathcal{S}}(u) = |\{ S \in \mathcal{S} : u \in S \}|$. Using this notation, a $t$-daisy $\mathcal{S}$ satisfies that every point $u$ outside its kernel has $\deg_{\mathcal{S}}(u) \leq t$.

The main parameters of a $t$-daisy $\mathcal{S} \subseteq 2^U$ are: (1) the *kernel size* $|K|$, (2) the *number of sets* $|\mathcal{S}|$, and (3) the *degree bound* $t$. As with sunflowers, we are typically interested in finding a large

[4] This is because the relaxed decoder can output $\perp$ in the majority of invocations and, say, output the wrong and correct values with equal (but small) probability.

daisy in a collection of subsets. More generally, since in our setting the collection of subsets will correspond to local views of a relaxed decoder, which are chosen according to some distribution, we wish to find a "heavy" daisy in a collection of *weighted* subsets.

We will need two lemmas: (1) a lemma that takes a weighted collection of subsets and extracts a $t$-daisy, for $t$ that is sublinear in $n$, which consists of heavy sets; and (2) a lemma that takes a $t$-daisy and extracts a simple daisy (i.e., 1-daisy), such that the union of its petals covers a large fraction of the domain.

Note that the former lemma yields a guarantee regarding the *weight* of the sets in a daisy, which corresponds to structure representing a relaxed local decoder according to its distribution of local views, whereas the latter lemma is a purely combinatorial lemma that "flattens out" the weights and gives a guarantee regarding the number of coordinates that are covered by a simple daisy that is derived from a $t$-daisy.

In the rest of this subsection, we will follow the approach in [FLV15] to derive the aforementioned lemmas, which will be instrumental to our approach.

**Extracting a heavy daisy from a weighted collection of subsets.** The following lemma shows that a sufficiently large collection of subsets, weighted according to a distribution, contains a daisy with a small kernel and heavy petals.

**Lemma 4.6** (daisy lemma)**.** *Let $\mathcal{T}$ be a collection of $cn$ subsets of $[n]$ of size $\ell$ each. Let $\mu$ be a distribution over $\mathcal{T}$. Then, for some $s \in [\ell]$, and $m = \max\{1, s-1\}$, there exists a $cn^{m/\ell}$-daisy $\mathcal{S} \subseteq \mathcal{T}$ with a kernel of size at most $\ell \cdot n^{1-s/\ell}$ and petals of size at most $s$, such that $\mu(\mathcal{S}) \geq 1/\ell$.*

We defer the proof of Lemma 4.6 to Appendix A.1. Note that in Lemma 4.6, the parameter $c$ is not necessarily a constant. However, when we actually invoke Lemma 4.6 it would be with respect to an absolute constant $c$.

**Extracting a simple daisy from a $t$-daisy.** The second lemma that we shall need shows that every $t$-daisy that covers a large part of the universe set contains a simple daisy (i.e., 1-daisy) of significant size.

**Lemma 4.7** (simple daisy lemma)**.** *Let $\mathcal{S}$ be a $t$-*daisy *with kernel $K$ and petals of size at most $s$, such that $|\bigcup_{T\in\mathcal{S}} T| = c'n$. Then, there exists a* simple *daisy $\mathcal{S}_0 \subseteq \mathcal{S}$ whose kernel is $K$, such that $|\mathcal{S}_0| \geq c'n - |K|$ if $s = 1$, and otherwise $|\mathcal{S}_0| \geq \frac{c'n-|K|}{ts^2}$.*

*Proof.* Initiate $\mathcal{S}_0$ to be the empty set. Our strategy will be to iteratively add *pairwise disjoint* sets to $\mathcal{S}_0$ until it satisfies the requirements of the lemma. All the sets we shall add to $\mathcal{S}_0$ are elements in the $t$-daisy $\mathcal{S}$, and therefore $\mathcal{S}_0 \subseteq \mathcal{S}$, which in turn implies that, for every $T \in \mathcal{S}_0$, it holds that $|T \setminus K| \leq t$. In the following, denote the subset of the domain that $\mathcal{S}$ covers by $M = \bigcup_{T\in\mathcal{S}} T$.

We first deal with the simple case where $s = 1$; that is, all petals $\{ T \setminus K : T \in \mathcal{S} \}$ of the daisy $\mathcal{S}$ contain just a single element. The idea is that here, for any point outside the kernel we can choose a single set that contains the point, obtaining a simple daisy.

In more detail, by definition of $M$, for every element $j \in M \setminus K$, which lies outside the kernel, there exists at least one petal of $\mathcal{S}$ that covers it, i.e., there exists $T \in \mathcal{S}$ such that $T \setminus K = \{ j \}$; choose such a set and add it to $\mathcal{S}_0$. By our construction of $\mathcal{S}_0$, for every distinct sets $T_1$ and $T_2$ in the daisy $\mathcal{S}_0$, it holds that the petals $T_1 \setminus K$ and $T_2 \setminus K$ are disjoint. Now, Since for every point

outside the kernel $j \in M \setminus K$, there exists at least one $T \in \mathcal{S}_0$, such that $T \setminus K = \{ j \}$, we have that $|\mathcal{S}_0| \geq c_2 n - |K|$, as required.

We now proceed to the other case, where $1 < s \leq \ell$. The idea here is that, as before, for any point outside the kernel we choose a single set that contains the point, only now, we remove all sets that intersect outside the kernel with the set that we chose, and proceed this way iteratively. Details follow.

For every $j \in M \setminus K$, there exists at least one $T \in \mathcal{S}$ such that $j \in T \setminus K$; choose such a set and add it to $\mathcal{S}_0$. In addition, remove $T$ from $\mathcal{S}$ and also every set $T^*$ in $\mathcal{S}$ such that $T \setminus K$ and $T^* \setminus K$ are not disjoint. We then reset $M$ to be the union of the sets in $\mathcal{S}$ after the sets were removed from it, and repeat the foregoing process until $M \setminus K$ is empty.

In order to lower bound the size of $\mathcal{S}_0$, we shall upper bound the loss in the cardinality of $M$ that happens in each iteration of removing sets from $\mathcal{S}$. Note that the the size of the union of all sets that are removed from $M$ in a single iteration is at most their number times their size, which is bounded by $ts^2$. This is the maximum loss in the cardinality of $M$, since we are over-counting by assuming that every element of a set $T$ that is added to $\mathcal{S}_0$ is contained in another removed set, and hence we can ignore the set $T$ during the computation. Consequently, we have that $|\mathcal{S}_0| \geq \frac{c'n - |K|}{ts^2}$. □

### 4.3 Construction of a global decoder

Recall that in Section 4.1 we obtained a *non-adaptive* $\ell'$-local relaxed decoder $D'$ for the code $C$ with soundness error $\varepsilon'$ and randomness complexity $r'$, where

$$\ell' = O(2^{\ell} \cdot \log(\ell)) \ , \ \varepsilon' = 1/\ell'^2 \ , \ r' = \log(n) + O(1) \ .$$

Since $D'$ is an $\ell'$-local *non-adaptive* decoder, the queries that it makes can be described by a distribution over $\ell'$-tuples of coordinates. More precisely, for every message location $i \in [k]$, there exist a predicate $f_i \colon \{0,1\}^{\ell'} \to \{0,1,\perp\}$ and distribution $\mu_i$ over size $\ell'$ subsets of $[k]$, such that $D'(i) = f_i(w|_I)$ for $I \sim \mu_i$.[5] Hereafter, we will identify the relaxed decoder $D'(i)$ with the predicate-distribution pair $(f_i, \mu_i)$.

Using the relaxed decoder $D'$, we construct a sample-based *global* decoder for the code $C$, which with high probability decodes the *entire* message of a *perfectly valid* codeword, using $O(n^{1-1/2\ell'^2})$ samples. Since it is information theoretically impossible to recover a $k$-bit message via $o(k)$ queries, this will imply that $n = \Omega(k^{2\ell'^2/(2\ell'^2-1)})$. (See Section 4.5 for a precise argument.)

Loosely speaking, the global decoder works as follows. First, it samples each coordinate independently with certain probability $p$ and tries to obtain local views of the relaxed local decoder $D'(i)$ for each location $i \in [k]$, while *reusing* the same samples.

Since the structure of the local views of $D'(i)$ does not guarantee that any local view will be captured in the aforementioned "binomial sampling" stage, the global decoder considers a subset of the local views of $D'$, which has the structure of a daisy; that is, it has a kernel of size that is smaller than the decoding radius such that outside of the kernel each point is covered by a small number of sets.

Our analysis will show that the binomial sampling stage is highly likely to yield petals of the daisy, but not its kernel, which is necessary to complete the petals into local views of the relaxed

[5] More accurately, for any $i \in [k]$ and query set $I$ chosen by the decoder $D'(i)$, there exist a predicate $f_{i,I} \colon \{0,1\}^{\ell'} \to \{0,1,\perp\}$. To simplify notation, since the query set $I$ will always be clear from the context, we write $f_i$ to refer to the corresponding $f_{i,I}$.

local decoder $D'$. To deal with that, the global decoder enumerates over all of the possible values of the kernel, and if one of the kernel assignments leads to a consensus of the decoding values for the corresponding local views, it outputs that value. (See Section 2 for a more detail high-level overview of the global decoder).

A precise description of the global decoder is given next.

**Construction 4.8.** Let $\{(f_i, \mu_i)\}_{i\in[k]}$ be the predicate-distribution pairs corresponding to the relaxed decoder $D'$ obtained in Section 4.1. For every $i \in [k]$, denote by $K_i$ and $\mathcal{P}_i$ the kernel and petals of the daisy $\mathcal{S}_i$ obtained by invoking Lemma 4.6 with respect to the support of $\mu_i$.

The global decoder $G$ receives query access to a string $w \in \{0,1\}^n$ and performs the following steps.

1. *Binomial sampling:* Set $p = n^{-1/2\ell'^2}$, and query each coordinate $j \in [n]$ independently with probability $p$. Denote by $Q$ the set of all coordinates that were queried. (If the size of $Q$ exceeds the query complexity, the global decoder can simply reject.)

2. *Local view generation:* For each $i \in [k]$, let $\mathcal{P}'_i \subseteq \mathcal{P}_i$ be the collection of all petals that were fully queried in the binomial sampling step (i.e., $\mathcal{P}'_i = \{\, P \in \mathcal{P}_i : P \subseteq Q \,\}$). For every petal $P \in \mathcal{P}'_i$, denote by $w|_P$ the restriction of the input $w$ to $P$.

3. *Global decoding:* for every $i \in [k]$, decode $x_i$ by performing the following steps for every assignment $\kappa \in \{0,1\}^{|K|_i}$ to the kernel.

   (a) For every fully-queried petal $P \in \mathcal{P}'_i$ and set $S \in \mathcal{S}_i$ that contains $P$, let $a_{S,\kappa}$ be the assignment to $S$ whose petal assignment is $w|_P$ and kernel assignment is $\kappa|_{S\setminus P}$. Let $A$ be the set of all such assignments.

   (b) If there exists $b \in \{0,1\}$ such that $f_i(a_{S,\kappa}) = b$ for every $a_{S,\kappa} \in A$, then output $b$ and proceed to decode $x_{i+1}$.

We proceed to analyse Construction 4.8 in Section 4.4.

### 4.4 Analysis of the construction

The following lemma shows that the construction described in Section 4.3 is a (global) decoder with sublinear query complexity (in the code's blocklength) that, with high probability, decodes the entire message encoded in a perfectly valid codeword.

**Lemma 4.9.** *Let $G$ be the algorithm defined in Construction 4.8. Given query access to a valid codeword $w = C(x)$ for $x \in \{0,1\}^k$, the algorithm $G$ makes $O(n^{1-1/2\ell'^2})$ queries to $w$ and satisfies that* $\Pr[G^w = x] \geq 2/3$.

*Proof.* Let $x \in \{0,1\}^k$, and denote $w = C(x)$. Let $Q$ be the set of all coordinates that the global decoder $G$ queried after sampling each element in $[n]$ with probability $p = n^{-1/2\ell'^2}$. Note that by standard binomial tail bounds, with probability at least $9/10$ the total query complexity of $G$ is $|G| = O(n^{1-1/2\ell'^2})$, as required (otherwise $G$ can simply abort). Suppose hereafter that this is the case.

Denote by $z \in \{0,1\}^k$ the output of $G$. We show that for every $i \in [k]$, the probability that $z_i \neq x_i$ is less than $1/10k$, and thus by a union bound $\Pr[G^w = x] \geq 9/10$. Thus, the total probability of success (including obtaining the desired query complexity) is $2/3$.

Fix $i \in [k]$. Recall that the global decoder $G$ decodes $x_i$ by considering a subset $\mathcal{S}_i$ of local views of the relaxed local decoder $D'(i)$, derived from the daisy lemma (Lemma 4.6). More precisely, during the binomial sampling stage $G$ queries the coordinates $Q$ and obtains a collection of fully queried petals of the daisy $\mathcal{S}_i$, which we denote by $\mathcal{P}'_i = \{ P \in \mathcal{P}_i : P \subseteq Q \}$.

However, for the global decoder $G$ to rule according to the relaxed local decoder $D'(i)$, it needs not only the fully-queried petals in $\mathcal{P}'_i$, but rather the complete local views of $D'(i)$ that contain these petals, i.e., the collection of subsets $\{ S \in \mathcal{S}_i : P \subseteq S \}$. To this end $G$ needs to obtain the value of the kernel $K_i$ of $\mathcal{S}_i$.

Since there is no guarantee that the kernel was fully queried (i.e., that $K_i \subseteq Q$), the global decoder $G$ enumerates over all possible assignments $\kappa \in \{0,1\}^{|K_i|}$ to the kernel, and considers the output of $D'(i)$ on each local view $S \in \mathcal{S}_i$ that consists of $w$ restricted to a petal $P \in \mathcal{P}'_i$ and value of $\kappa$.

Recall that for every fully-queried petal $P \in \mathcal{P}'$ and set $S \in \mathcal{S}_i$ that contains $P$, we denote by $a_{S,\kappa}$ the assignment to $S$ whose petal assignment is $w|_P$ and kernel assignment is $\kappa|_{S \setminus P}$. Observe that, by definition (see Construction 4.8), the global decoder $G$ outputs the correct value $x_i$ if the set of fully-queried petals $\mathcal{P}'_i$ is non empty, and the following conditions hold:

1. There exists a kernel assignment $\kappa^*$ such that for *every* fully-queried petal $P \in \mathcal{P}'_i$ and set $S \in \mathcal{S}_i$ that contains $P$, the relaxed local decoder $D'(i)$ outputs the correct value $x_i$ given the local view $a_{S,\kappa^*}$ (i.e. $f_i(a_{S,\kappa^*}) = x_i$).

2. For any other kernel assignment $\kappa \in \{0,1\}^{|K_i|}$, *there exists* a fully-queried petal $P \in \mathcal{P}'_i$ and set $S \in \mathcal{S}_i$ that contains $P$, such that $D(i)$ outputs the correct value $x_i$ or aborts given the local view $a_{S,\kappa}$ (i.e. $f_i(a_{S,\kappa}) \in \{ x_i, \perp \}$).

Note that the first item above guarantees that at least one kernel assignment will lead the global decoder $G$ to output the correct value $x_i$, whereas the second item guarantees that with high probability no kernel assignment will lead $G$ to output the incorrect value $\neg x_i$.

The next two claims establish that the foregoing conditions are satisfied and that $\mathcal{P}'_i$ is non-empty with high probability. In the following, recall that $\ell' = O(1)$, and by Lemma 4.6, there exists $s \in [\ell']$ and $m = \max\{1, s-1\}$ such that, denoting $t = cn^{m/\ell'}$, the collection $\mathcal{S}_i$ is a $t$-daisy with a kernel $K_i$ of size at most $\ell' \cdot n^{1-s/\ell'}$ and $\mu_i(\mathcal{S}_i) \geq 1/\ell'$.

**Claim 4.10.** *There exists a kernel assignment $\kappa^* \in \{0,1\}^{|K_i|}$ such that the assignment $a_{S,\kappa^*}$, for every $S \in \mathcal{S}_i$ containing a queried petal $P \in \mathcal{P}'_i$, satisfies $f_i(a_{S,\kappa^*}) = x_i$. Furthermore, $\mathcal{P}'_i$ is non-empty with probability at least $1 - \frac{1}{10k}$.*

*Proof.* The existence of the desired kernel assignment is straightforward, we begin by showing it, and then proceed to the furthermore clause of the claim, which is the main part of the proof.

Set $\kappa^*$ to be the kernel assignment that coincides with $w$; that is $\kappa = w|_{K_i}$. In this case, by definition, for every $S \in \mathcal{S}_i$ the assignment $a_{S,\kappa^*}$ equals to the local view $w|_S$ of the relaxed local decoder $D'(i)$, for a valid codeword $w = C(x)$.

By the perfect completeness of $D'$, it holds that given query access to $w = C(x)$, any local view of $D'$ leads to outputting $x_i$, and thus $f_i(a_{S,\kappa^*}) = x_i$ for all $S \in \mathcal{S}_i$. It remains to show that with high probability there exists a least one petal of $\mathcal{S}_i$ that was queried in the binomial sampling stage.

To this end, we next argue that not only $\mathcal{S}_i$ has large density, but that it also covers a large fraction of the domain $[n]$. Recall that the weight that the local decoder $D'(i)$ gives $\mathcal{S}_i$ is larger than the soundness error of $D'(i)$, i.e., $\mu_i(\mathcal{S}_i) \geq 1/\ell' \geq \varepsilon'$. Thus, the fractional size of the set of all

elements covered by $\mathcal{S}_i$ must be larger than the decoding radius $\delta$ (otherwise, replacing $\mathcal{S}_i$ with the values of a codeword that disagrees with $x_i$ will leave $w$ within the decoding radius, and thus break the soundness condition), i.e., $|\cup_{S\in\mathcal{S}_i} S| > \delta n$.

Now, we can invoke Lemma 4.7 to "pluck" intersecting petals in the $t$-daisy $\mathcal{S}_i$, for $t = cn^{m/\ell'}$, where $m = \max\{1, s-1\}$, and derive a subset that is a simple daisy (a 1-daisy). Namely, Lemma 4.7 implies that there exists a *simple* daisy $\mathcal{S}_i^* \subseteq \mathcal{S}_i$ whose kernel is $K_i$ (same as $\mathcal{S}_i$), such that

$$|\mathcal{S}_i^*| \geq \frac{\delta n - |K_i|}{ts^2} \geq \frac{\delta n - \ell' \cdot n^{1-s/\ell'}}{cn^{m/\ell'}s^2} = \Omega\left(n^{1-(m/\ell')}\right) \ .$$

Denote the set of petals of $\mathcal{S}_i^*$ by $\mathcal{P}_i^*$, and note that $|\mathcal{P}_i^*| \geq (|\cup_{S\in\mathcal{S}_i^*} S| - |K_i|)/s = \Omega(n^{1-(m/\ell')})$.

Since the petals of a simple daisy are *disjoint*, observe that for any petal $P \in \mathcal{P}_i^*$, the probability of querying all $s$ elements of $P$ during the binomial sampling step is $p^s = n^{-s/2\ell'^2}$. Since $\mathcal{S}_i^*$ contains $d \coloneqq \Omega(n^{1-(m/\ell')})$ pairwise disjoint petals, the probability that no petal of $\mathcal{S}_i^*$ was queried is

$$\Pr[\mathcal{P}_i^* = \emptyset] = (1-p^s)^d = \left(1 - \frac{1}{n^{s/(2\ell'^2)}}\right)^{\Omega(n^{1-(m/\ell')})} \leq e^{\Omega\left(-n^{1-\frac{m}{\ell'}-\frac{s}{2\ell'^2}}\right)} \leq \frac{1}{10k} \ ,$$

and so $\mathcal{P}_i'$ is non-empty with probability at least $1 - 1/(10k)$, concluding the proof of Claim 4.10. □

**Claim 4.11.** *For every kernel assignment $\kappa \in \{0,1\}^{|K_i|}$, with probability at least $1 - \frac{2^{-|K_i|}}{10k}$, there exists a queried petal $P \in \mathcal{P}_i'$ and $S \in \mathcal{S}_i$ containing $P$ such that $f_i(a_{S,\kappa}) \in \{x_i, \perp\}$.*

*Proof.* Let $\kappa \in \{0,1\}^{|K_i|}$ be a kernel assignment. By Lemma 4.6, the kernel $K_i$ of the $t$-daisy $\mathcal{S}_i$ satisfies $|K_i| \leq \ell' \cdot n^{1-s/\ell'}$. In particular, note that the fractional size of the kernel is smaller than the decoding radius $\delta$.

Recall that the global decoder $G$ gets access to a perfectly valid codeword $w = C(x)$, and emulates query access to a string $z$ that agrees with $w$ outside of the kernel and with $\kappa$ inside the kernel (i.e., $z_j = w_j$ for every $j \in [n] \setminus K_i$, and $z|_{K_i} = \kappa$). Since $|K_i| \leq \delta n$, we have that $z$ is within the decoding radius of the relaxed local decoder $D'$.

By the relaxed decoding condition of $D'$, it holds that given query access to $z$, with probability at least $1 - \varepsilon'$, the local view of $z$ chosen by $D'(i)$ will lead to either outputting the correct value $x_i$ or the abort symbol $\perp$; more precisely,

$$\Pr_{I\sim\mu_i}\left[f_i(z|_I) \in \{x_i, \perp\}\right] \geq 1 - \varepsilon' = 1 - \frac{1}{\ell'^2} \ . \tag{4.1}$$

Recall that the global decoder relies on a subset (a $t$-daisy) $\mathcal{S}_i$ of the local views of $D'(i)$ to perform the decoding. We argue that $\mathcal{S}_i$ contains a high density (according to $\mu_i$) set of local views that correspond to local views that will lead $D'(i)$ to either outputting the correct value $x_i$ or the abort symbol $\perp$. To this end, let $\mathcal{G}_i \subseteq \mathcal{S}_i$ be the subset of all "good" $S \in \mathcal{S}_i$ such that $f_i(z|_S) \in \{x_i, \perp\}$. Note that $\mathcal{G}_i$ is also a $t$-daisy with respect to the kernel $K_i$, where $t = cn^{m/\ell'}$.

By Lemma 4.6, we have that $\mu_i(\mathcal{S}_i) \geq 1/\ell'$. Thus, Eq. (4.1) implies that

$$\mu_i(\mathcal{G}_i) \geq \frac{1}{\ell'} - \frac{1}{\ell'^2} \geq \varepsilon' \ . \tag{4.2}$$

Similarly to the argument Claim 4.10, we observe that not only $\mathcal{G}_i$ has large density, but that it also covers a large fraction of the domain $[n]$. More accurately, by Eq. (4.2) we have that the density of $\mathcal{G}_i$ is larger than the soundness error $\varepsilon'$, and so $|\cup_{S\in\mathcal{G}_i} S| > \delta n$.

To see the above, observe that taking any valid codeword $\alpha$ that decodes to $x_i$ and replacing only the contents of the daisy (of density larger than the soundness error $\varepsilon'$) with the content of a valid codeword $\beta$ that decodes to $\neg x_i$, results in a "hybrid" word $\gamma$ that: (1) decodes to $\neg x_i$ with probability larger than $\varepsilon'$ (because according to the construction, with probability larger than $\varepsilon'$ the decoder's local view consists of values from $\beta$); and (2) if the daisy does not cover a large fraction of the domain (more than the decoding radius), then it is still within the decoding radius of $\alpha$, which means that, by the second condition of relaxed LDC, it should decode to $\{x_i, \perp\}$ with probability at least $1 - \varepsilon'$. In conclusion, if the daisy has density larger than $\varepsilon'$, then it must cover a large fraction (the decoding radius) of the domain.

We conclude the proof of the claim by showing that with probability at least $1 - \frac{2^{-|K_i|}}{10k}$ there exists a least one petal of $\mathcal{G}_i$ that was fully queried in the binomial sampling stage, via a similar strategy as in Claim 4.10. To this end, we invoke Lemma 4.7 to obtain a *simple* daisy $\mathcal{G}_i^* \subseteq \mathcal{G}_i$ whose kernel is $K_i$ (same as the kernel of $\mathcal{G}_i$ and $\mathcal{S}_i$), such that $|\mathcal{G}_i^*| \geq \Omega\left(n^{1-(m/\ell')}\right)$, where $m = s - 1$.[6] Denote the set of petals of $\mathcal{G}_i^*$ by $\mathcal{P}^*$, and note that.

$$|\mathcal{P}_i^*| = \frac{|\mathcal{G}_i^*| - |K_i|}{s} = \Omega\left(n^{1-(m/\ell')}\right) \ .$$

Since the petals of a simple daisy are *disjoint*, observe that for any petal $P \in \mathcal{P}_i^*$, the probability of querying all $s$ elements of $P$ during the binomial sampling step is $p^s = n^{-s/2\ell'^2}$. Since $\mathcal{G}_i^*$ contains $d \coloneqq \Omega(n^{1-(m/\ell')})$ pairwise disjoint petals and $|K_i| \leq \ell' \cdot n^{1-s/\ell'}$, the probability that no petal of $\mathcal{G}_i^*$ was queried is

$$\Pr[\mathcal{P}_i^* = \emptyset] = e^{\Omega\left(-n^{1-\frac{s-1}{\ell'}-\frac{s}{2\ell'^2}}\right)} \leq \frac{2^{-|K_i|}}{10k} \ ,$$

which proves Claim 4.11. □

Wrapping up the argument, for any $i \in [k]$, by Claim 4.10, *there exists* a kernel assignment for $K_i$ such that with probability $1 - 1/(10k)$ there is a fully queried petal $P \in \mathcal{P}_i'$ that leads $D'(i)$ to output the correct value (i.e., there exists $S \in \mathcal{S}_i$ that contains the fully queried petal $P$ such that $f_i(a_{S,\kappa^*}) = x_i$).

Furthermore, for any $i \in [k]$, by Claim 4.11 we have that *for any* kernel assignment for $K_i$, with probability at least $1 - \frac{2^{-|K_i|}}{10k}$, there is a fully queried petal $P \in \mathcal{P}_i'$ that leads $D'(i)$ to either output the correct value or abort (i.e., there exists $S \in \mathcal{S}_i$ that contains the fully queried petal $P$ such that $f_i(a_{S,\kappa^*}) \in \{ x_i, \perp \}$). Taking a union bound over all possible kernel assignments $\kappa \in \{0,1\}^{|K_i|}$, we get that with probability at least $1 - 1/(10k)$, there is a set of fully queried petals $\{ P_i \in \mathcal{P}_i' \}_{i \in [k]}$ such that for every $i \in [k]$, the petal $P_i$ leads $D'(i)$ to either output the correct value or abort.

We thus showed that for every $i \in [k]$, the probability that the global decoder $G$ fails to decode the message bit $x_i$ is at most $1/10k$. Finally, taking another union bound over all decoding indices $i \in [k]$, we obtain that the foregoing holds for all $i \in [k]$ *simultaneously* with probability at least $9/10$. This concludes the proof of Lemma 4.9. □

[6] Note that here we use the fact that the simple daisy lemma (Lemma 4.7) has a stronger conclusion for the special case where $s = 1$.

### 4.5 Deriving the lower bound

Recall that we have started with an $\ell$-local relaxed LDC $C\colon \{0,1\}^k \to \{0,1\}^n$ with a constant decoding radius $\delta$ and constant locality $\ell$, and that we wish to show that the blocklength of $C$ satisfies $n = \Omega\left(k^{1+\alpha}\right)$, where $\alpha = \alpha(\ell, \delta)$ is a constant.

So far, we have shown the there exists a global decoder $G$ for the code $C$, which with probability $2/3$ decodes the *entire* message of a *perfectly valid* codeword, using $O(n^{1-1/2\ell'^2})$ samples, where $\ell' = O(2^\ell \cdot \log(\ell))$. The following simple claim shows that decoding an entire codeword requires a least a number of queries that is linear in the dimension of the code.

**Claim 4.12.** *Let $C : \{0,1\}^k \to \{0,1\}^n$ be a code. If there exists a randomised algorithm $\mathcal{A}$ that makes $q$ queries to a codeword $C(x)$, for some $x \in \{0,1\}^k$, such that $\Pr\left[\mathcal{A}^{C(x)} = x\right] \geq 2/3$, then $q \geq k$.*

*Proof.* Suppose towards contradiction that the number of queries that $\mathcal{A}$ makes is at most $k-1$. We use (the easy direction of) Yao's minimax principle to show that this implies that $\mathcal{A}$ returns the wrong answer with probability at least $1/2$, in contradiction to the claim's hypothesis. To this end, it suffices to show that there exists a distribution $\mathcal{D}$ over $n$-bit strings on which every deterministic algorithm that makes at most $k-1$ queries errs with probability at least $1/2$.

The distribution $\mathcal{D}$ is defined by simply selecting uniformly at random a message $x \in \{0,1\}^k$ and outputting $C(x)$. Let $\mathcal{B}$ be a *deterministic* algorithm that receives an input $w$ drawn from $\mathcal{D}$ and makes at most $k-1$ queries to $w$.

Let $I$ be the set of queries that the deterministic algorithm makes. Note that $I$ is deterministically fixed and $|I| \leq k-1$. After querying the indices in $I$, the algorithm $B$ can be described by a (deterministic) mapping $f$ from $2^{|I|}$ to $2^k$, which maps the local view $w|_I$ to a $k$-bit message. Since $|I| \leq k-1$, the range of $f$ is of size at most $2^{k-1}$, and so it contains at most half of the possible values of $x$. Thus, with probability at least $1/2$ the input $C(x)$ drawn from $\mathcal{D}$ corresponds to an $x \in \{0,1\}^k$ that is not in the range of $f$, and hence $\mathcal{B}$ errs. □

Applying Claim 4.12 with respect to the global decoder $G$ implies that $n = \Omega(k^{2\ell'^2/(2\ell'^2-1)})$, which concludes the proof of Theorem 1.

## Acknowledgements

We are grateful to Oded Goldreich for numerous insightful comments and suggestions that significantly improved the exposition of this paper. We also thank Arnab Bhattacharyya and Sivakanth Gopi for a helpful discussion regarding LDC and LCC lower bounds. We thank Noga Ron-Zewi for extended discussions on constructions of relaxed LDCs. We thank Marcel De Sena Dall'Agnol for helpful comments regarding earlier versions of this manuscript.

## References

[BDG17] Jop Briët, Zeev Dvir, and Sivakanth Gopi. Outlaw distributions and locally decodable codes. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 20:1–20:19, 2017.

[BDSS11] Arnab Bhattacharyya, Zeev Dvir, Amir Shpilka, and Shubhangi Saraf. Tight lower bounds for 2-query LCCs over finite fields. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 638–647, 2011.

[BDYW11] Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 519–528, 2011.

[BGGZ18] Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, 2018.

[BGH+04] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, 2004.

[BGT16] Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query LCCs over large alphabet. *arXiv preprint arXiv:1611.06980*, 2016.

[CG18] Clément L. Canonne and Tom Gur. An adaptivity hierarchy theorem for property testing. *Computational Complexity*, 27(4):671–716, 2018.

[CGdW09] Victor Chen, Elena Grigorescu, and Ronald de Wolf. Efficient and error-correcting data structures for membership and polynomial evaluation. *arXiv preprint arXiv:0909.3696*, 2009.

[CGS20] Alessandro Chiesa, Tom Gur, and Igor Shinkar. Relaxed locally correctable codes with nearly-linear block length and constant query complexity. *31st ACM-SIAM Symposium on Discrete Algorithms (to appear)*, 2020.

[DH13] Irit Dinur and Prahladh Harsha. Composition of low-error 2-query pcps using decodable pcps. *SIAM J. Comput.*, 42(6):2452–2486, 2013.

[DJK+02] Amit Deshpande, Rahul Jain, Telikepalli Kavitha, Satyanarayana V Lokam, and Jaikumar Radhakrishnan. Better lower bounds for locally decodable codes. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 184–193. IEEE, 2002.

[DSW17] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Superquadratic lower bound for 3-query locally correctable codes over the reals. *Theory of Computing*, 13(1):1–36, 2017.

[Efr12] Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, 41(6):1694–1703, 2012.

[FLV15] Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, 2015.

[GG16] Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:192, 2016.

[GG18] Oded Goldreich and Tom Gur. Universal locally testable codes. *Chicago J. Theor. Comput. Sci.*, 2018.

[GGK15] Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, 2015.

[GKST02] Oded Goldreich, Howard Karloff, Leonard J Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 175–183, 2002.

[Gol04] Oded Goldreich. Short locally testable codes and proofs. In *ECCC (later appeared in Property Testing 2010)*, 2004.

[Gol17] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.

[GR17] Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 39:1–39:43, 2017.

[GR18] Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, 27(1):99–207, 2018.

[GRR18] Tom Gur, Govind Ramnarayan, and Ron D. Rothblum. Relaxed locally correctable codes. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 27:1–27:11, 2018.

[GS06] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM (JACM)*, 53(4):558–655, 2006.

[GS10] Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *Computational Complexity*, 19(1), 2010.

[HSX+12] Cheng Huang, Huseyin Simitci, Yikang Xu, Aaron Ogus, Brad Calder, Parikshit Gopalan, Jin Li, and Sergey Yekhanin. Erasure coding in Windows Azure storage. In *Presented as part of the 2012 USENIX Annual Technical Conference*, pages 15–26, 2012.

[KdW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Sciences*, 69(3):395–420, 2004.

[KS17] Swastik Kopparty and Shubhangi Saraf. Local testing and decoding of high-rate error-correcting codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:126, 2017.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, 2000.

[KV10] Tali Kaufman and Michael Viderman. Locally testable vs. locally decodable codes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 670–682. Springer, 2010.

[Mei09] Or Meir. Combinatorial construction of locally testable codes. *SIAM Journal on Computing*, 39(2):491–544, 2009.

[MR10] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5):29:1–29:29, 2010.

[Oba02] Kenji Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 39–50. Springer, 2002.

[RR19] Noga Ron-Zewi and Ron Rothblum. Local proofs approaching the witness length. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:127, 2019.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 2004.

[Vid13] Michael Viderman. Strong ltcs with inverse poly-log rate and constant soundness. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 330–339. IEEE, 2013.

[WdW05] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *International Colloquium on Automata, Languages, and Programming*, pages 1424–1436. Springer, 2005.

[Woo12] David P. Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. *Journal of Computer Science and Technology*, 27(4):678–686, 2012.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1):1:1–1:16, 2008.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.

# A Deferred proofs

We provide the proofs of lemmas and claims that were deferred from Section 4.

## A.1 Proof of the daisy lemma

Let $\mathcal{T}$ be a collection of $cn$ subsets of $[n]$ of size $\ell$ each. Let $\mu$ be a distribution over $2^{[n]}$, whose support is $\mathcal{T}$. We show that for some $s \in [\ell]$, and $m = \max\{1, s-1\}$, there exists a $cn^{m/\ell}$-daisy $\mathcal{S} \subseteq \mathcal{T}$ with a kernel of size at most $\ell \cdot n^{1-s/\ell}$ and petals of size at most $s$, such that $\mu(\mathcal{S}) \geq 1/\ell$.

Our high-level strategy consists of two steps: (1) we first iteratively construct a sequence of daisies $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell \subseteq 2^{[n]}$ with kernels $K_1, K_2, \ldots, K_\ell \subseteq [n]$, respectively; and (2) we then argue that there exists $s \in [\ell]$ for which the daisy $\mathcal{S}_s$ with respect to kernel $K_s$ satisfies the desired conditions. We construct this sequence of daisies as follows.

**Construction A.1.** *Given a collection $\mathcal{T}$ of $cn$ subsets of $[n]$ of size $\ell$ each, we construct a sequence of daisies $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell \in 2^{[n]}$ with kernels $K_1, K_2, \ldots, K_\ell \subseteq [n]$ as follows.*

1. *Set $\mathcal{T}_1 = \mathcal{T}$.*
2. *Perform the following steps iteratively, for $i \in [\ell]$:*
   (a) *let $K_i$ be the set of all $j \in [n]$ such that $\deg_{\mathcal{T}_i}(j) > cn^{i/\ell}$.*
   (b) *let $\mathcal{S}_i$ be the family of all subsets $T \in \mathcal{T}_i$ such that $|T \setminus K_i| \leq i$.*
   (c) *let $\mathcal{T}_{i+1}$ be $\mathcal{T}_i \setminus \mathcal{S}_i$.*

We proceed to show three structural claims regarding the daisies in the sequence $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell$; Namely: (1) bounding the sizes of their kernels, (2) showing their union covers the original collection $\mathcal{T}$, and (3) bounding the number of sets that contain each point outside of their kernels. Subsequently, we will show that at least one of these daisies satisfies all requirements of the lemma.

We begin with the following claim, which shows that for each $i \in [\ell]$, the daisy $\mathcal{S}_i$ has a sufficiently small kernel.

**Claim A.2.** *For every $i \in [\ell]$, it holds that $|K_i| < \ell n^{1-i/\ell}$.*

*Proof.* By Item 2c of Construction A.1, for every $i \in [\ell]$, it holds that $\mathcal{T}_i \subseteq \mathcal{T}_1$. Hence, as $|\mathcal{T}_1| = cn$, we know that $|\mathcal{T}_i| \leq cn$, for every $i \in [\ell]$.

Fix $i \in [\ell]$. By Item 2a of Construction A.1,

$$\sum_{j \in K_i} \deg_{\mathcal{T}_i}(j) > |K_i| cn^{i/\ell} \tag{A.1}$$

Since all the subsets of $[n]$ have cardinality $\ell$,

$$\sum_{j \in K_i} \deg_{\mathcal{T}_i}(j) \leq \ell |\mathcal{T}_i| \leq \ell cn \tag{A.2}$$

The claim follows from Eqs. (A.1) and (A.2). □

The next claim shows that the union of all subsets in all the daisies $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell$ covers the original collection of subsets $\mathcal{T}$ (equivalently, $\mathcal{T}_1$).

**Claim A.3.** $\bigcup_{i \in [\ell]} \mathcal{S}_i = \mathcal{T}_1$.

*Proof.* Note that for $i = \ell$, the condition in Item 2b of Construction A.1 lets $\mathcal{S}_\ell$ be the family of all subsets $T \in \mathcal{T}_\ell$ such that $|T \setminus K_\ell| \leq \ell$. The cardinality of each set in $\mathcal{T}_\ell$ is $\ell$, and hence, trivially, $\mathcal{S}_\ell = \mathcal{T}_\ell$, which in turn means that every set in $\mathcal{T}_1$ is in one of the families in $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell\}$ and the claim follows. □

Next, we show a claim which shows that for each $i \in [\ell]$, every point outside of the kernel of $\mathcal{S}_i$ is incident in only a small number of sets of $\mathcal{S}_i$.

**Claim A.4.** *For every $i \in [\ell]$, and $j \in [n] \setminus K_i$, $\deg_{\mathcal{S}_i}(j) \leq cn^{\frac{\max\{1,(i-1)\}}{\ell}}$.*

*Proof.* For $i = 1$, the claim follows directly from Item 2a of Construction A.1.

Suppose towards contradiction that there exists $i \in \{2, 3, \ldots, \ell\}$ and $j \in [n] \setminus K_i$, such that $\deg_{\mathcal{S}_i}(j) > cn^{(i-1)/\ell}$. Let $T$ be a set in $\mathcal{S}_i$ such that $j \in T$. Let $T'$ be the subset of $T$ that consists of $T \cap K_i$ and every index $h \in T \setminus K_i$ such that $\deg_{\mathcal{S}_i}(h) > cn^{(i-1)/\ell}$.

The size of $T'$ is at least $|T \cap K_i| + 1$, because we assumed $T'$ has an element in $T \setminus K_i$. Hence, $|T'| \geq \ell - (i-1)$. We next show that

$$\deg_{\mathcal{S}_{i-1}}(h) > cn^{\frac{(i-1)}{\ell}} \ ,$$

for every $h \in T'$. This implies that $T' \subseteq K_{i-1}$ and in turn that $|T \setminus K_{i-1}| = \ell - |T'| \leq i - 1$. Thus, $T \in \mathcal{S}_{i-1}$ and consequently, by Items 2c and 2b of Construction A.1, $T$ is not in $\mathcal{S}_i$, in contradiction to our initial assumption.

By Item 2a of Construction A.1, for every $h \in T' \cap K_i$, it holds that $\deg_{\mathcal{T}_i}(h) > cn^{i/\ell}$, which in turn implies that

$$\deg_{\mathcal{T}_{i-1}}(h) > cn^{i/\ell} > cn^{(i-1)/\ell} \ ,$$

because $\mathcal{T}_i \subseteq \mathcal{T}_{i-1}$, by Item 2c of Construction A.1.

By item 2b, $\mathcal{S}_i \subseteq \mathcal{T}_i$, and we already deduced that $\mathcal{T}_i \subseteq \mathcal{T}_{i-1}$. Thus, $\mathcal{S}_i \subseteq \mathcal{T}_{i-1}$ and therefore, for every $h \in T' \setminus K_i$ if $\deg_{\mathcal{S}_i}(h) > cn^{(i-1)/\ell}$, then also $\deg_{\mathcal{T}_{i-1}}(h) > cn^{(i-1)/\ell}$, and the claim follows. □

Finally, we rely on Claim A.2, Claim A.3, and Claim A.4 to show that at least one of the daisies in in the sequence $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell$ satisfies all of the conditions of the lemma.

**Claim A.5.** *For some $s \in [\ell]$, $\mathcal{S}_s$ is a $cn^{\max\{1,(s-1)\}/\ell}$-daisy with a kernel $K$, such that $\mu(\mathcal{S}_s) \geq 1/\ell$, $|K| < \ell \cdot n^{1-s/\ell}$, and $|T \setminus K| \leq s$, for every $T \in \mathcal{S}_s$.*

*Proof.* By Claim A.3, the sequence of daisies $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\ell$ covers $\mathcal{T}_1$, i.e., $\bigcup_{j \in [\ell]} \mathcal{S}_j = \mathcal{T}_1$. Therefore, there exists $s \in [\ell]$ such that $\mu(\mathcal{S}_s) \geq 1/\ell$. We take $K$ to be the set $K_s$. According to the construction of $K_s$, for every $T \in \mathcal{S}_s$ it holds that $|T \setminus K_s| \leq s$, and by Claim A.2, we have that $|K_s| < \ell n^{1-s/\ell}$. Finally, by Claim A.4, $\deg_{\mathcal{S}_s}(j) \leq cn^{\max\{1,(s-1)\}/\ell}$, for every $j \in [n] \setminus K_s$. □

This concludes the proof of Lemma 4.6.

## A.2 Proof of Claim 4.2

Let $D$ be an adaptive $\ell$-local relaxed decoder for the code $C \colon \{0,1\}^k \to \{0,1\}^n$. We show that $C$ also has a *non-adaptive* $2^\ell$-local relaxed decoder with the same decoding radius.

Fix $i \in [k]$. Note that the adaptive decoder $D$ can be viewed as a distribution over binary decision trees; that is, $D(i)$ first tosses coins to obtain a random string $\rho$ which determines the

binary decision tree that $D$ uses deterministically to determine the output $D(i)$ given query access to input $w$.

Recall that every non-leaf vertex of a decision tree is labelled by an index in $[n]$, and one of the edges leaving it towards a child is labelled 0 and the other is labelled 1. The relaxed local decoder $D$ uses a decision tree, which it selects at random, by starting from the root of the tree, reading its label $i$ and querying $w_i$. It then proceeds to the child of the root with the edge corresponding to the value of $x_i$. The child is treated in the same manner as the root if it is an internal vertex, and if it is a leaf, its label takes value in $\{0, 1, \perp\}$, which is the output of $D(i)$.

Let $D'$ be an algorithm that operates as follows. Given explicit input $i \in [k]$, it first tosses coins, exactly like $D(i)$, to obtain a random string $\rho$ which determines the binary decision tree that $D(i)$ uses. Then, it queries *all* of the indices labelling the vertices of the decision tree (corresponding to all possible queries that $D(i)$ might have been given any possible input $w$). Finally, $D'$ uses the query values to compute the answer $D(i)$ will have returned and returns it. Note that there are $2^\ell$ such labels.

The choice of queries of $D'$ depends only on the decision tree chosen, hence it is non-adaptive. The query complexity of $D'$ is $2^\ell$ and it returns the exact same answer as $D$ will on the same input and random coin tosses. This concludes the proof of Claim 4.2.

### A.3 Proof of Claim 4.3

Let $D$ be an $\ell$-local relaxed decoder for the code $C\colon \{0,1\}^k \to \{0,1\}^n$, which errs with probability at most $1/3$. We use $D$ to construct an $O(\ell \cdot \log(1/\varepsilon))$-local relaxed decoder $D'$ that errs with probability at most $\epsilon$ and has perfect completeness.

On input $i \in [k]$ and query access to a string $w \in \{0,1\}^n$, the relaxed local decoder $D'$ operates as follows:

1. Invoke $\log(1/\varepsilon)$ parallel executions of $D$ with the same input parameters;
2. When the invoked executions attempt to query $w$, the relaxed local decoder $D'$ collects all of the queries and asks them simultaneously, returning the values of the queries to the executions that requested them;
3. Finally, $D'$ collects all the outputs from the executions and returns $b \in \{0,1\}$ if all the executions returned $b$, and otherwise returns $\perp$.

Note that all the queries are used in a non-adaptive manner, and that there are at most $\ell \cdot \log(1/\varepsilon)$ of them.

By the completeness of the original relaxed local decoder $D$, if $w$ is a valid codeword, then $b = x_i$ with probability 1. Hence, $D'$ has perfect completeness. On the other hand, if $w$ is $\delta$-close to $C$, then by the relaxed decoder condition of relaxed LDC, it holds that $x \notin \{x_i, \perp\}$ with probability less than $(1/3)^{\log(1/\varepsilon)} < \varepsilon$. Thus, $D'$ has the desired amplified soundness. This concludes the proof of Claim 4.3.

### A.4 Proof of Claim 4.4

Let $D$ be a $\ell$-local relaxed decoder for a *binary* code $C$ with $\ell = O(1)$, randomness complexity $r$, and constant error probability $\varepsilon$. We use $D$ to construct a relaxed decoder $D'$ with the same parameters as $D$, except it has locality $O(\ell)$ and randomness complexity $\log(n) + O(1)$.

Fix $i \in [k]$. Consider a $2^r \times 2^n$ matrix where the rows correspond to all possible random strings $\gamma$ used by the relaxed local decoder and the columns correspond to all inputs $w \in \{0,1\}^n$ that are within the decoding radius of $C$. The entry $(\gamma, w)$ of the matrix corresponds to the output of $D^w(i;\gamma)$, that is, the output of the relaxed local decoder when given query access to $w$ and random coins $\gamma$.

Note that for every codeword $w = C(x)$, by the perfect completeness of $D(i)$ the value of each entry in a $w$ column equals the correct message value $x_i$. By the relaxed decoding condition of $D(i)$, for each $w$ column that is within the decoding radius of $C$, at least $1 - \varepsilon$ fraction of the entries are in $\{ x_i, \perp \}$.

We show that there exists a multi-set, $S$, of size $O(n)$ of the rows such that the every column $w$ restricted to $S$ has at most $O(\varepsilon)$ fraction of entries taking the wrong value $\neg x_i$. Thus, we obtain a relaxed local decoder $D'$ that uses only $\log_2 |S| = \log n + O(1)$ random coins, by simply running the original decoder $D$ but with respect to random coins selected uniformly from $S$ (rather than from $\{0,1\}^r$). To obtain soundness error $\varepsilon$ we use $O(1)$ parallel repetitions.

We use the probabilistic method to show the existence of a small multi-set $S$ as above. Consider a multi-set $S$ of the rows, of size $t$, chosen uniformly at random and fix input $w$. By the Chernoff bound, with probability $2^{-\Omega(t)}$ over the choice of $S$, at most $O(\varepsilon)$ fraction of entries of $w$ restricted to $S$ take the wrong value $\neg x_i$. Thus, by setting $t = \log(2^n) + O(1)$ and applying the union bound, we obtain that there exists a multi-set $S$ as desired. Since the new relaxed local decoder selects at random from $S$, it can be implemented using $\log_2 t + O(1)$ random coins. This concludes the proof of Claim 4.4.